Scientific Visualization and Parallel I/O at Extreme Scale

Rob Ross Tom Peterka Rob Latham
Mathematics and Computer Science Division
Argonne National Laboratory
rross@mcs.anl.gov, tpeterka@mcs.anl.gov

Han-Wei Shen Yuan Hong
Department of Computer Science and Engineering
The Ohio State University
hwshen@cse.ohio-state.edu, hongy@cse.ohio-state.edu

Kwan-Liu Ma Chaoli Wang
Department of Computer Science
University of California at Davis
ma@cs.ucdavis.edu, chawang@ucdavis.edu

Hongfeng Yu
Combustion Research Facility
Sandia National Laboratories
hfyu@ucdavis.edu

Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

UChicago Argonne LLC

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

UltraVis SciDAC Institute for Ultrascale Visualization

SDM CENTER

Large-scale data sets

Application teams are beginning to generate 10s of Tbytes of data in a single simulation. For example, a recent GTC run on 29K processors on the XT4 generated over 54 Tbytes of data in a 24 hour period [1].

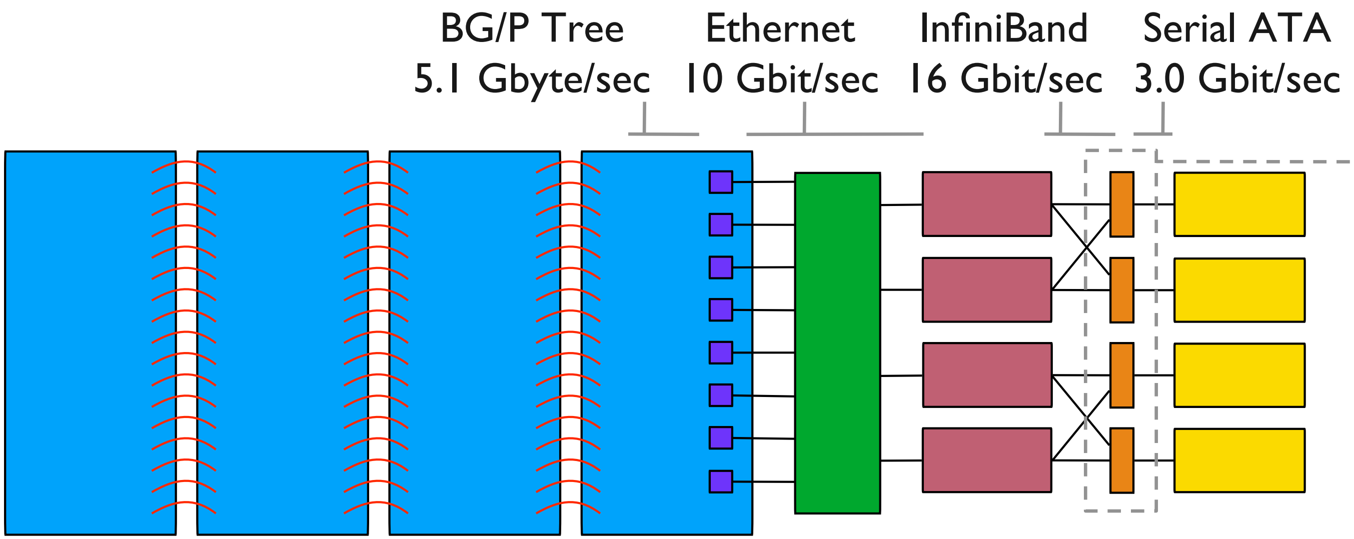
Similarly, the FLASH team running at 16-32K cores on BG/P is generating 74 Gbyte checkpoints every 3 hours and 16 Gbyte plotfiles every 10-15 minutes [2].

Data requirements for select 2008 INCITE applications at ALCF

PI	Project	On-Line Data	Off-Line Data
Lamb, Don	FLASH: Buoyancy-Driven Turbulent Nuclear Burning	75TB	300TB
Fischer, Paul	Reactor Core Hydrodynamics	2TB	5TB
Dean, David	Computational Nuclear Structure	4TB	40TB
Baker, David	Computational Protein Structure	1TB	2TB
Worley, Patrick H.	Performance Evaluation and Analysis	1TB	1TB
Wolverton, Christopher	Kinetics and Thermodynamics of Metal and Complex Hydride Nanoparticles	5TB	100TB
Washington, Warren	Climate Science	10TB	345TB
Tsigelny, Igor	Parkinson's Disease	2.5TB	50TB
Tang, William	Plasma Microturbulence	2TB	10TB
Sugar, Robert	Lattice QCD	1TB	44TB
Siegel, Andrew	Thermal Striping in Sodium Cooled Reactors	4TB	8TB
Roux, Benoit	Gating Mechanisms of Membrane Proteins	10TB	10TB

[1] S. Klasky, personal correspondence, June 19, 2008.
[2] K. Riley, personal correspondence, July 15, 2008.

Blue Gene/P parallel storage system



HW bottleneck is here. Controllers can manage only 4.6 Gbyte/sec.

Peak I/O system bandwidth is 78.2 Gbyte/sec.

Gateway nodes run parallel file system client software and forward I/O operations from HPC clients

640 Quad core PowerPC 450 nodes with 4 Gbytes of RAM each

Commodity network primarily carries storage traffic

900+ port 10 Gigabit Ethernet Myricom switch complex

Storage nodes run parallel file system software and manage incoming FS traffic from gateway nodes

136 two dual core Opteron servers with 8 Gbytes of RAM each

Enterprise storage controllers and large racks of disks connected via InfiniBand or Fibre Channel

17 DataDirect S2A9900 controller pairs with 480 1 Tbyte drives and 8 InfiniBand ports per pair

Architectural diagram of 557 TF Argonne Leadership Computing Facility Blue Gene/P I/O system.

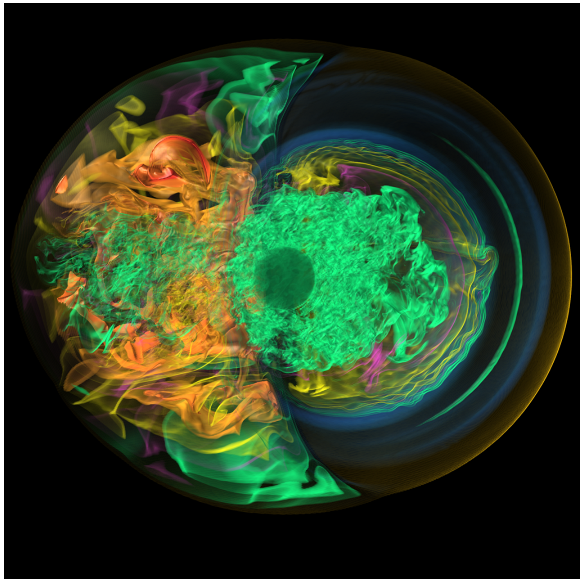
Analyzing large-scale data sets

Where should data analysis be performed? Options include: using a separate analysis resource (e.g. cluster), using the large-scale compute system itself, or (possibly) using advanced functionality within the storage system.

Should we process the data first? With knowledge of the underlying I/O system, access patterns, and data organization, data can be stored to make reading “easier” for the storage system.

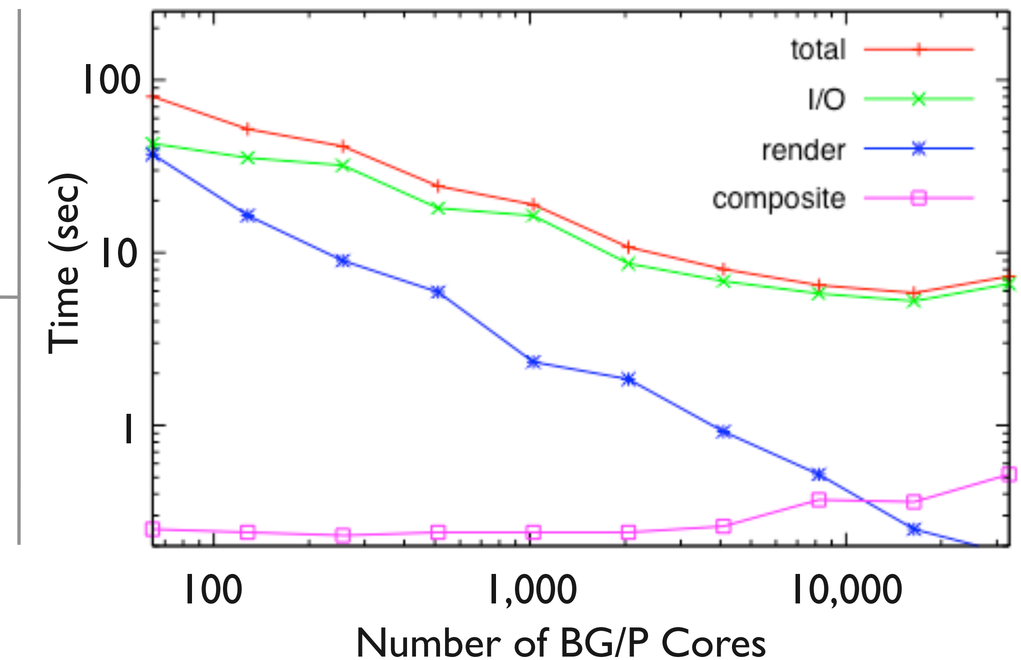
Can the amount of I/O be reduced? Certain algorithms can incorporate techniques for reducing I/O (e.g. early ray termination in volume rendering). Alternatively, we could perform some analysis while data is still in memory.

Visual data analysis on leadership systems



As data sizes grow, I/O access begins to dominate run time, and the value of special-purpose processors such as GPUs is diminished. In these cases, it might make more sense to perform analysis on leadership systems, rather than making the significant investment in networking infrastructure necessary to enable high performance I/O to a separate “visualization cluster”.

Rendering of 1120^3 astrophysics time step from the VH-1 hydrodynamics code was performed on the ALCF Blue Gene/P system, generating a 1600^2 image. Looking at the time spent in analysis, rendering time is significant, but **I/O time clearly dominates.**

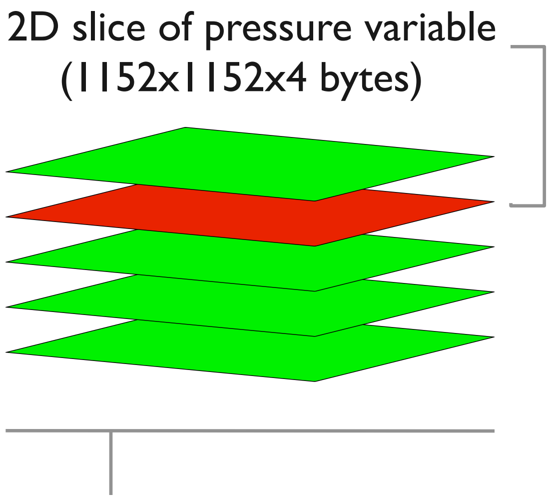


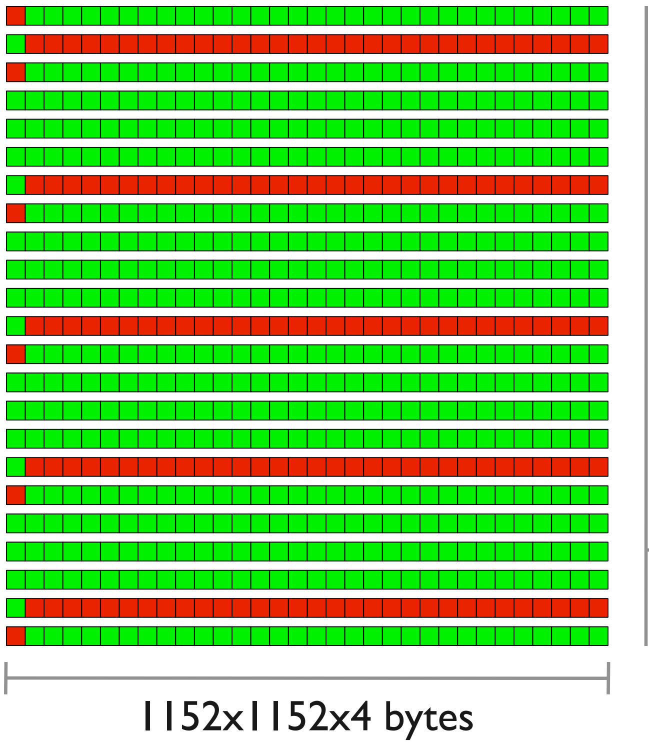
Thanks to J. Blondin (NCSU) and A. Mezzacappa (ORNL) for providing the sample data set.

Peterka, T., Yu, H., Ross, R., Ma, K.-L., “Parallel volume rendering on the IBM Blue Gene/P”, Proc. of EGPGV’08, April 2008.

Understanding data organization

When data is stored during a simulation, analysis is rarely taken into consideration. Data might be stored in a file per process, or in one large file with interleaved variables. When analysis is finally performed, the resulting I/O accesses may not be optimal. If we are going to analyze the data repeatedly, it may make sense to reorganize the data before we begin analysis.





The astrophysics data described previously is stored as a netCDF data set with 5 large 3D variables. The netCDF software interleaves 2D slices of variables in the file.

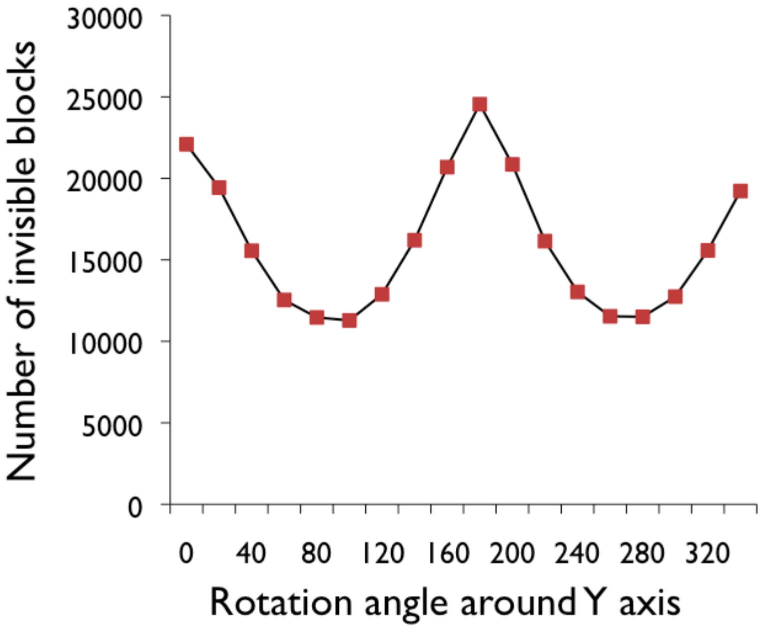
In the netCDF file, a small header (first red block) results in these slices being stored at a slight offset relative to their width. The long red regions represent slices of a single variable.

To generate an image using a single variable, only the header and every fifth slice needed. **Reading these directly from the output data set doubles the I/O time**.

Reducing data access

Visual analysis techniques can often avoid reading data by determining either a priori, or at run time, that certain regions will not be visible in the resulting image. This can significantly reduce the amount of data accessed, but it may limit the degree to which regions may be processed in parallel.

Y. Hong and H.-W. Shen have been working with one of the Visible Human datasets from the National Library of Medicine. The magnetic resonance data is 512x512x1728 with single 12-bit value at each point (stored in two bytes). Data is partitioned into 16^3 voxels (110,592 total).



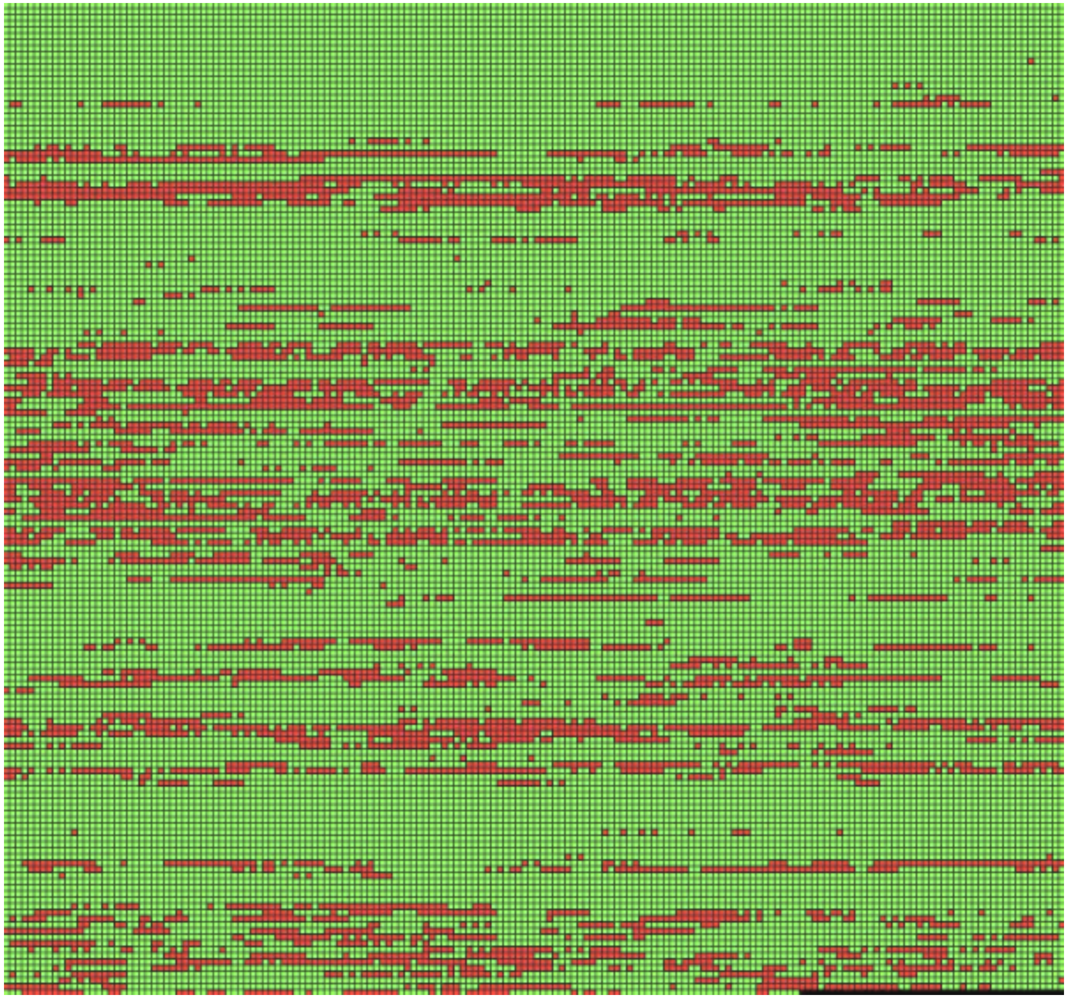
When run time methods are used to reduce data access, only a fraction of the total data set must be read. The graph shows the number of invisible blocks for a set of views rotated around the Y axis. The troughs represent “head on” and “back on” views, where more surface is visible.

The Visible Human Project. http://www.nlm.nih.gov/research/visible/

Access patterns during volume rendering

When this type of optimization is applied, data accesses often appear random to the storage system. While much less data is accessed, optimizations such as read-ahead within the file system might be disabled due to the irregularity of incoming reads. Further, the underlying hard drives are likely to spend significant time seeking from one region to another.

This image shows a 2D row-major representation of data blocks from the Visible Human dataset as stored on one server. Volume rendering was performed on this dataset using an out-of-core algorithm that only reads visible data blocks. Only the red blocks were accessed.



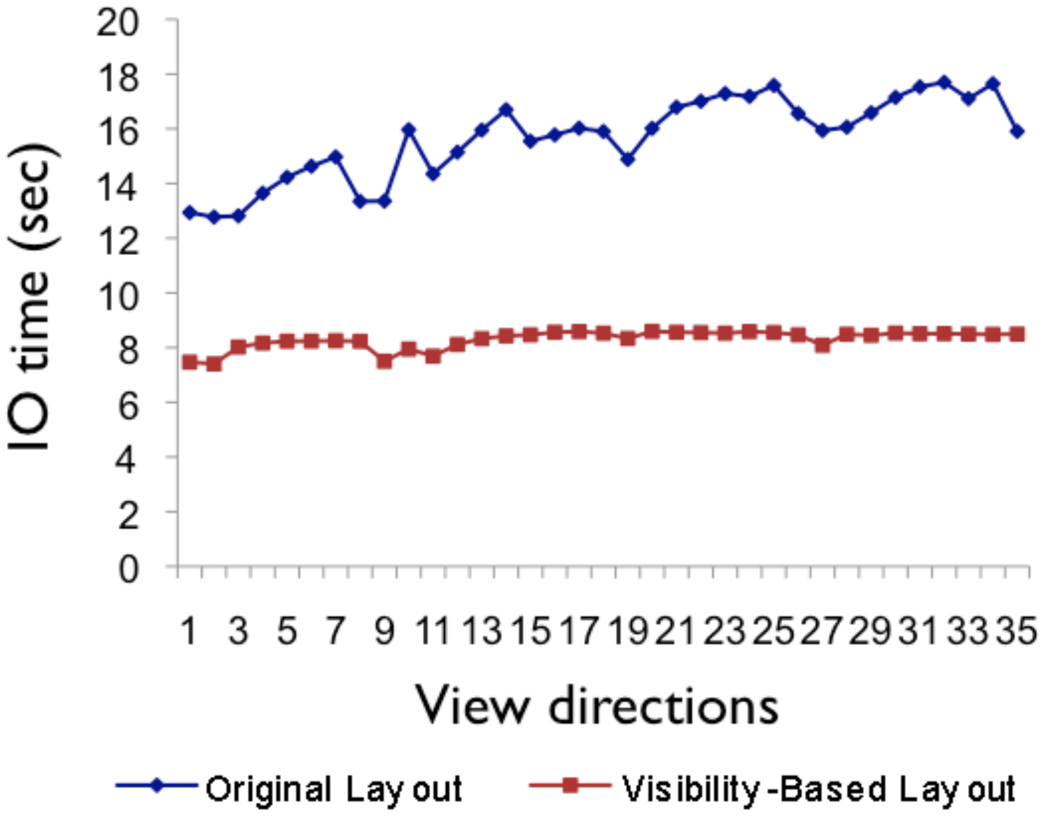
Y. Hong and H.-W. Shen, "Histogram-based visibility culling in visualizing large volume data", OSU Technical Report OSU-CISRC-7/08-TR38, 2008.

Organizing data sets for efficient access

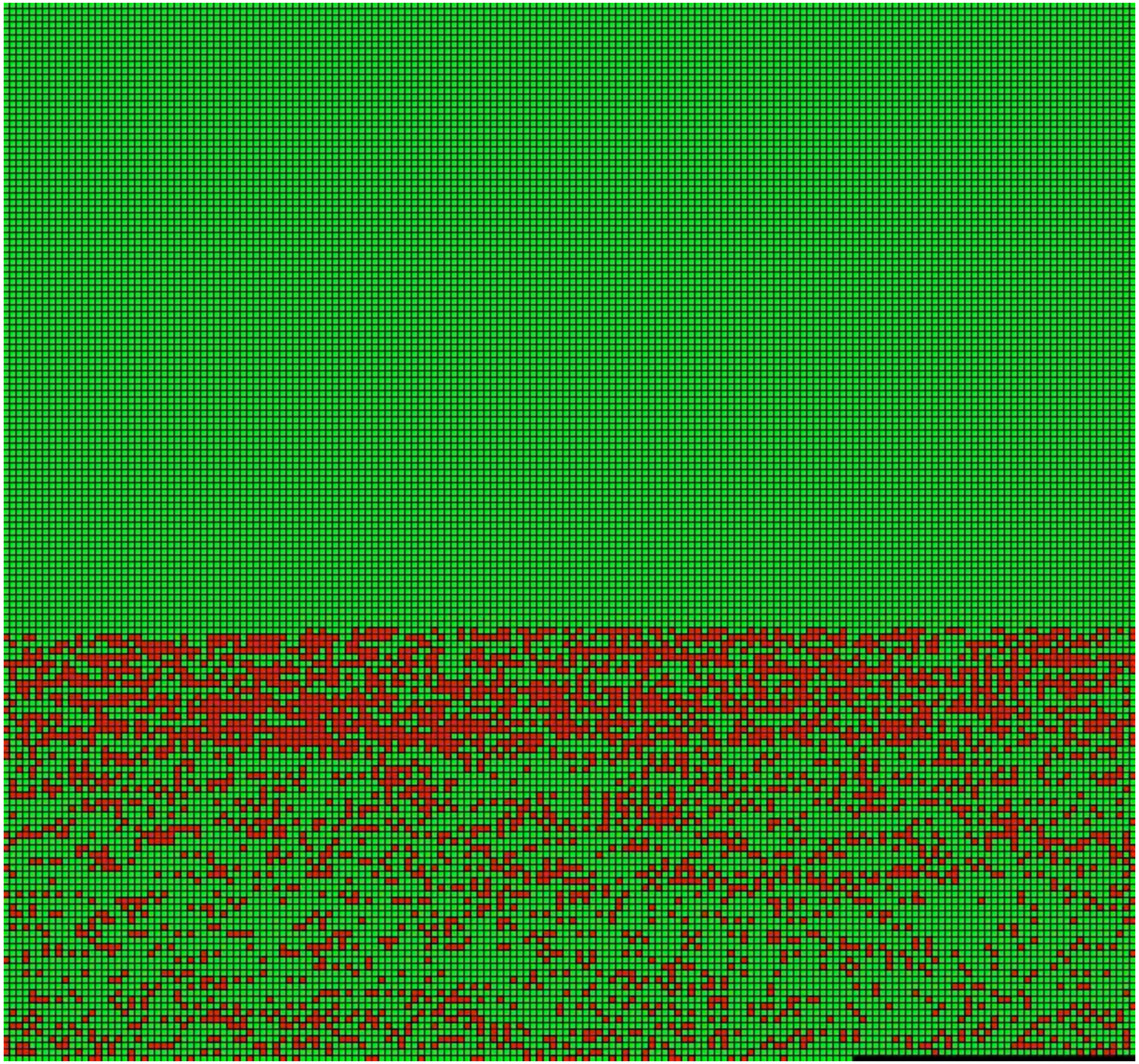
A **visibility feature vector** is an *n*-dimensional tuple that serves a measure of the visibility of a block from a variety of view directions, independent of transfer function. By calculating these vectors for each block of a data set, visibility of a given block can be determined with a high degree of accuracy, without re-reading the blocks. We can further use these vectors to cluster blocks with similar access characteristics into adjacent regions in the file.

Comparison of access times with original file layout and visibility-based layout from a variety of view directions using a fixed transfer function.

Data gathered on IBM BG/L at ANL using 64 compute nodes, 16 server PVFS.



Access pattern for visibility-based layout



Accesses for a typical viewpoint and transfer function fall into less than 50% of the blocks stored on a single server.

Data gathered on IBM BG/L at ANL using 64 compute nodes, 16 server PVFS.

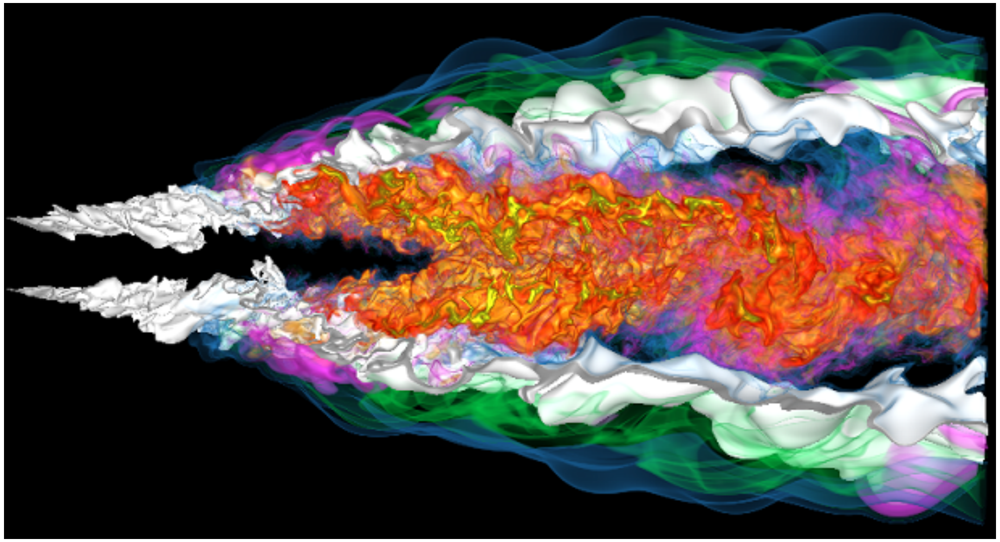
In situ analysis and data reduction

In situ analysis incorporates analysis routines into the simulation code. This technique allows analysis routines to operate on data while it is still in memory, potentially significantly reducing the I/O demands.

One way to take advantage of in situ techniques is to perform initial analysis for the purposes of data reduction. With help from the application scientist to identify features of interest, we can compress data of less interest to the scientist, reducing I/O demands during simulation and further analysis steps.

The feature of interest in this case is the mixture fraction with an iso value of 0.2 (white surface). Colored regions are a volume rendering of the HO2 variable (data courtesy J. Chen (SNL)).

By compressing data more aggressively the further it is from this surface, we can attain a compression ratio of 20-30x while still retaining full fidelity in the vicinity of the surface.



C. Wang, H. Yu, and K.-L. Ma, "Application-driven compression for visualizing large-scale time-varying volume data", IEEE Computer Graphics and Applications, accepted for publication.